Refine Search

Search Results -

Terms	Documents
L2 and (internet OR web OR www)	7

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L3

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Monday, March 21, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L3	L2 and (internet OR web OR www)	7	L3
L2	L1 and applet	8	L2
L1	717/177.ccls.	89	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L2 aND (installation ADJ applet)	0

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L4

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Monday, March 21, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L4	L2 aND (installation ADJ applet)	0	L4
L3	L2 aND applet	11	L3
L2	FTP AND (download ADJ file) and (html ADJ document)	21	L2
L1	FTP AND (download near file) and (html ADJ document)	35	L1

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 11 of 11 returned.

☐ 1. Document ID: US 6701374 B2

L3: Entry 1 of 11 File: USPT Mar 2, 2004

US-PAT-NO: 6701374
DOCUMENT-IDENTIFIER: US 6701374 B2

TITLE: Method and apparatus for dynamic proxy insertion in network traffic flow

DATE-ISSUED: March 2, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gupta; Amit	Fremont	CA		
Baehr; Geoffrey	Menlo Park	CA		

US-CL-CURRENT: 709/238; 370/235, 370/236, 709/237, 709/242

ABSTRACT:

In embodiments of the invention, a method and apparatus for dynamic proxy insertion in network traffic path is described. According to one or more embodiments of the invention, a request and/or response message may be modified to include one or more thru-proxy tags to identify a network (or traffic) node (e.g., a proxy, server, or intermediary). For example, a request directed to a server or a response directed to a client may be altered to insert a plurality of intermediate or final destination designations. In so doing, a path of a request or response may be altered dynamically. A thru-proxy tag in a response may be inserted in a related request to identify a destination or node such that the request is sent to the destination in the thru-proxy tag before being sent to an origin server. Thru-proxy tags may be used to identify multiple and/or alternate destinations.

3 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☑ 2. Document ID: US 6671723 B2

L3: Entry 2 of 11 File: USPT Dec 30, 2003

US-PAT-NO: 6671723

DOCUMENT-IDENTIFIER: US 6671723 B2

TITLE: Method and apparatus for scanning a web site in a distributed data processing system for problem determination

DATE-ISSUED: December 30, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Andrew Quoc Anh	Austin	TX		
Turek; John Joseph Edward	South Nyack	NY		
Shtalhaim; Menachem	Haifa			IL

US-CL-CURRENT: 709/224; 370/242

ABSTRACT:

A method and apparatus for identifying problems associated with a web site. A scan of a web site is initiated by a plurality of agents, wherein each of the plurality of agents are at a different location in the distributed data processing system. Results of the scan are obtained from the plurality of agents. The results of the scan are analyzed to determine if a problem is associated with the web site.

5 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

☑ 3. Document ID: US 6615259 B1

L3: Entry 3 of 11 File: USPT Sep 2, 2003

US-PAT-NO: 6615259
DOCUMENT-IDENTIFIER: US 6615259 B1

TITLE: Method and apparatus for scanning a web site in a distributed data processing system for problem determination

DATE-ISSUED: September 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Andrew Quoc Anh	Austin	TX		
Turek; John Joseph Edward	South Nyack	NY		
Shtalhaim; Menachem	Haifa			IL

US-CL-CURRENT: 709/224; 370/242

ABSTRACT:

A method and apparatus for identifying problems associated with a web site. A scan of a web site is initiated by a plurality of agents, wherein each of the plurality of agents are at a different location in the distributed data processing system. Results of the scan are obtained from the plurality of agents. The results of the scan are analyzed to determine if a problem is associated with the web site.

35 Claims, 11 Drawing figures
Exemplary Claim Number: 33
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 4. Document ID: US 6567857 B1

L3: Entry 4 of 11 File: USPT May 20, 2003

US-PAT-NO: 6567857
DOCUMENT-IDENTIFIER: US 6567857 B1

TITLE: Method and apparatus for dynamic proxy insertion in network traffic flow

DATE-ISSUED: May 20, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gupta; Amit	Fremont	CA		
Baehr; Geoffrey	Menlo Park	CA		

US-CL-CURRENT: 709/238; 370/235, 709/242

ABSTRACT:

In embodiments of the invention, a method and apparatus for dynamic proxy insertion in network traffic path is described. According to one or more embodiments of the invention, a request and/or response message may be modified to include one or more thru-proxy tags to identify a network (or traffic) node (e.g., a proxy, server, or intermediary). For example, a request directed to a server or a response directed to a client may be altered to insert a plurality of intermediate or final destination designations. In so doing, a path of a request or response may be altered dynamically. A thru-proxy tag in a response may be inserted in a related request to identify a destination or node such that the request is sent to the destination in the thru-proxy tag before being sent to an origin server. Thru-proxy tags may be used to identify multiple and/or alternate destinations.

7 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 5. Document ID: US 6118768 A

L3: Entry 5 of 11 File: USPT Sep 12, 2000

US-PAT-NO: 6118768
DOCUMENT-IDENTIFIER: US 6118768 A

TITLE: Apparatus and methods for use therein for an ISDN LAN modem utilizing browser-based configuration with adaptation of network parameters

DATE-ISSUED: September 12, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhatia; Rajiv	Marlboro	NJ		
Zhang; Siuling C.	Stony Brook	NY		

US-CL-CURRENT: 370/254; 370/469, 709/222, 709/250

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem that is suited for small user environments and which contains an internal ISDN router having a self-contained network hub for inter-connecting multiple network devices, such as workstations, to each other through a local area network (LAN) and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. Advantageously, to facilitate and simplify its configuration, the LAN modem automatically adapts itself to a current network environment of a workstation connected thereto, via the LAN, and then communicates with that workstation through a browser executing thereat to obtain configuration information from a user situated at the workstation.

27 Claims, 50 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 6. Document ID: US 6108330 A

L3: Entry 6 of 11 File: USPT Aug 22, 2000

US-PAT-NO: 6108330
DOCUMENT-IDENTIFIER: US 6108330 A

TITLE: Apparatus and methods for use therein for an ISDN LAN modem that selects among a plurality of DNS servers for responding to a DNS query

DATE-ISSUED: August 22, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

Bhatia; Rajiv	Marlboro	NJ
Zhang; Siuling C.	Stony Brook	NY

US-CL-CURRENT: 370/352; 370/401

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem (300) (and various aspects thereof) that is particularly, though not exclusively, suited for small user environments and which contains an internal ISDN router (305) having a self-contained network hub (340) for inter-connecting multiple network devices, such as workstations (10), to each other through a local area network (LAN) and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. Advantageously, to facilitate and simplify its configuration, the LAN modem automatically adapts itself to a current network environment of a workstation connected thereto, via the LAN, and then communicates with that workstation through a browser executing thereat to obtain configuration information from a user situated at the workstation. Additionally, the LAN modem, through use of a multi-tiered routing hierarchy including both destination- and source-based routing, accommodates several modalities of network communication not heretofore possible in a conventional router. Specifically, several different workstations can simultaneously communicate through the LAN modem with a common remote network (60) and share a single user account at a corresponding network service provider.

23 Claims, 51 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

☑ 7. Document ID: US 6094659 A

L3: Entry 7 of 11 File: USPT Jul 25, 2000

US-PAT-NO: 6094659
DOCUMENT-IDENTIFIER: US 6094659 A
**** See image for Certificate of Correction ****

TITLE: Web server for use in a LAN modem

DATE-ISSUED: July 25, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhatia; Rajiv	Marlboro	NJ		

US-CL-CURRENT: 707/104.1; 379/93.14, 707/10

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem that is suited for small user environments and which contains an internal ISDN router

having a self-contained network hub for inter-connecting multiple network devices, such as workstations, to each other through a local area network and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. The LAN modem includes an internal web server for autonomously constructing and downloading a web page, through dynamic selective insertion of predefined event-specific web page components into a web page template, to the workstation. The resulting page informs a user stationed at the workstation of a failure condition or other operational event that then occurred at the LAN modem. The specific page components inserted into the template are selected based on the particular failure condition or other operational event which occurred.

30 Claims, 50 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

☐ 8. Document ID: US 6052803 A

L3: Entry 8 of 11 File: USPT Apr 18, 2000

US-PAT-NO: 6052803
DOCUMENT-IDENTIFIER: US 6052803 A

TITLE: Key-based technique for assuring and maintaining integrity of firmware stored in both volatile and non-volatile memory

DATE-ISSUED: April 18, 2000

INVENTOR-INFORMATION:				
NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhatia; Rajiv	Marlboro	NJ		
Elhindi; Tayseer M.	Ocean	NJ		
Sun; Matthew	Holmdel	NJ		

US-CL-CURRENT: 714/49

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem (300) (and various aspects thereof) that is particularly, though not exclusively, suited for small user environments and which contains an internal ISDN router (305) having a self-contained network hub (340) for inter-connecting multiple network devices, such as workstations (10), to each other through a local area network (LAN) and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. Advantageously, to facilitate and simplify its configuration, the LAN modem automatically adapts itself to a current network environment of a workstation connected thereto, via the LAN, and then communicates with that workstation through a browser executing thereat to obtain configuration information from a user situated at the workstation. Additionally, the LAN modem, through use of a multi-tiered routing hierarchy including both destination- and source-based routing, accommodates several modalities of network communication not heretofore possible in a conventional router. Specifically,

several different workstations can simultaneously communicate through the LAN modem with a common remote network (60) and share a single user account at a corresponding network service provider. Also, the LAN modem can simultaneously route packet traffic between multiple workstations on the LAN and different remote networks (60, 70) through different ISDN connections simultaneously existing between the LAN modem and corresponding network service providers.

25 Claims, 50 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

☐ 9. Document ID: US 6029203 A

L3: Entry 9 of 11 File: USPT Feb 22, 2000

US-PAT-NO: 6029203
DOCUMENT-IDENTIFIER: US 6029203 A

TITLE: Apparatus and methods for use therein for an ISDN LAN modem that provides enhanced network activity

DATE-ISSUED: February 22, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhatia; Rajiv	Marlboro	NJ		
Douglas; C. Paul	Matawan	NJ		
Zhang; Siuling C.	Stony Brook	NY		

US-CL-CURRENT: 709/244; 709/217, 709/248

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem that is suited for small user environments and which contains an internal ISDN router having a self-contained network hub for inter-connecting multiple network devices, such as workstations, to each other through a local area network (LAN) and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. Through use of a multi-tiered routing hierarchy including both destination- and source-based routing, the LAN modem accommodates several modalities of network communication not heretofore possible in a conventional router. Specifically, several different workstations can simultaneously communicate through the LAN modem with a common remote network and share a single user account at a corresponding network service provider. Also, the LAN modem can simultaneously route packet traffic between multiple workstations on the LAN and different remote networks through different ISDN connections simultaneously existing between the LAN modem and corresponding network service providers.

44 Claims, 50 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

☐ 10. Document ID: US 6028848 A

L3: Entry 10 of 11 File: USPT Feb 22, 2000

US-PAT-NO: 6028848
DOCUMENT-IDENTIFIER: US 6028848 A

TITLE: Apparatus and methods for use therein for an ISDN LAN modem utilizing internal DNS and DHCP servers for transparent translation of local host names to IP addresses

DATE-ISSUED: February 22, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhatia; Rajiv	Marlboro	NJ		
Elhindi; Tayseer M.	Ocean	NJ		
Zhang; Siuling C.	Stony Brook	NY		

US-CL-CURRENT: 370/257; 370/401, 370/475

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem (300) (and various aspects thereof) that is particularly, though not exclusively, suited for small user environments and which contains an internal ISDN router (305) having a self-contained network hub (340) for inter-connecting multiple network devices, such as workstations (10), to each other through a local area network (LAN) and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. Advantageously, to facilitate and simplify its configuration, the LAN modem automatically adapts itself to a current network environment of a workstation connected thereto, via the LAN, and then communicates with that workstation through a browser executing thereat to obtain configuration information from a user situated at the workstation. Additionally, the LAN modem, through use of a multi-tiered routing hierarchy including both destination- and source-based routing, accommodates several modalities of network communication not heretofore possible in a conventional router. Specifically, several different workstations can simultaneously communicate through the LAN modem with a common remote network (60) and share a single user account at a corresponding network service provider.

28 Claims, 59 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

☐ 11. Document ID: US 6023724 A

L3: Entry 11 of 11 File: USPT Feb 8, 2000

US-PAT-NO: 6023724
DOCUMENT-IDENTIFIER: US 6023724 A

TITLE: Apparatus and methods for use therein for an ISDN LAN modem that displays fault information to local hosts through interception of host DNS request messages

DATE-ISSUED: February 8, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhatia; Rajiv	Marlboro	NJ		
Stypko; Gregory H.	Jackson	NJ		
Zhang; Siuling C.	Stony Brook	NY		

US-CL-CURRENT: 709/218; 709/219, 709/224, 709/225, 709/237

ABSTRACT:

Apparatus, and accompanying methods for use therein, for an ISDN LAN modem that is suited for small user environments and which contains an internal ISDN router having a self-contained network hub for inter-connecting multiple network devices, such as workstations, to each other through a local area network and for permitting each of those devices to each gain access through the router to any one of a number of different remote networks. The LAN modem communicates network failure messages to a host workstation connected to the LAN by intercepting and responding to various DNS (domain name system) messages issued by that workstation and intended for a remote DNS server. Specifically, the LAN modem supplies its own network (IP) address in response to these messages, thus assuming a role of both a remote DNS server and a remote web server in order to implement a mechanism through which a fault-specific web page can be dynamically constructed and downloaded to the workstation for subsequent display, through a browser executing thereat. The page, once rendered, provides a specific message pertinent to the failure.

20 Claims, 50 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 41

Terms	Documents
L2 aND applet	11

Previous Page Next Page Go to Doc#

Google Web Images Groups News Froogle Local[New!] more »
JAR applet installer Search Advanced Search Preferences

Web

Results 1 - 10 of about 16,300 for JAR applet installer. (0.18 seconds)

Vitalsecurity.org - We're Calm like a Bomb: Firefox Spyware ...
... the Java Runtime Environment, the initial **installer** taking the ... used can recognise and install the **applet**, then it ... of you at the back, here's the .**Jar** file in ...
www.vitalsecurity.org/2005/ 03/firefox-spyware-infects-ie.html - 101k - Mar 19, 2005 - Cached - Similar pages

JNLP Tags - Quick Reference
... optional, Describes the documentbase for the **applet** as a ... class attribute if the first **jar** in the ... extension, Describes a component or **installer** extension that is ...
www.vamphq.com/jnlpquick.html - 29k - Cached - Similar pages

JNLP Tag Reference
... file as an application update - only updated **jar** files will. ... security?, resources*, (application-desc | **applet**-desc | component-desc | **installer**-desc) ...
www.vamphq.com/jnlpref.html - 83k - Cached - Similar pages
[More results from www.vamphq.com]

Web **Installer**
... If you wish to sign the **applet** with your ... Once signed, place the WebInst.**jar** and WebInst.cab ... InstallAnywhere/resource/Web folder, and rebuild your **installers**. ...
www.zerog.com/iamanual/usermanual_ia55/ WebHelp/distribution/web_**installer**.htm - 7k - Cached - Similar pages

Signing **Applets** for Java Plugin
... Security System and Using PVCj As Your **Installer** for the ... you have to somehow get the PVCj **jar** file onto ... modify your HTML page that starts the **applet** to point ...
www.suitable.com/docs/signingsignplug.html - 16k - Cached - Similar pages

JNI : Java Glossary
... The **installer Applet** will copy the Worker.**jar** to C:\program files\Netscape\Communicator\ program\java\classes or perhaps D:. Thereafter, you could run it ...
www.mindprod.com/jgloss/jni.html - 44k - Cached - Similar pages

Deploying Java Extensions in Java Plug-in
... The Java **installer** must be bundled as a **JAR** file, and the resulting **JAR** file must be specified as Implementation-URL in the **applet's** MANIFEST file. ...
java.sun.com/products/plugin/1.3/docs/extensions.html - 24k - Cached - Similar pages

JavaFoil - the **Applet**
... javafoil.**jar** and mhclasses.**jar**) into this directory. ... You can run the **applet** by loading this html page into ... you can also download a Windows **Installer** file which ...
www.mh-aerotools.de/airfoils/jf_**applet**.htm - 12k - Cached - Similar pages

Installing Ptplot
... The InstallShield **installer** places the ptplot , histogram and pxgraph startup scripts in ... Ptplot as an **applet** is to copy the appropriate ***applet.jar** file into ...
www.cs.bham.ac.uk/resources/ums/java/ packages/ptplot3.1/ptolemy/plot/doc/install.htm - 10k - Cached - Similar pages

Install Autodesk MapGuide Viewer, Java Edition Release 6.5
... Choose "Uninstall" from the **applet installer**. OR; Open the folder: [Your User Home directory] > Library > Java > Extensions. Drag the mgjava.**jar** file to the ...
navigatela.lacity.org/download/Distribution/Install/ - 22k - Cached - Similar pages

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

JAR applet installer Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Installing Ptplot

Ptplot is distributed in several formats.

- The first format is the usual tar file and zip file format.
- The second format is an experimental format using the Installshield JShield installer.

Ptplot Installation under Unix or Windows with Cygwin from tar or zip files

Ptplot is shipped with the java files precompiled, so you need not recompile. However, if you are using the standalone scripts such as `ptplot`, then you may want to go through the installation procedure so that the standalone scripts are properly localized.

If you are using Windows and you want to rebuild Ptplot or localize the standalone scripts, then you will need to install the Cygwin toolkit from `http://sourceware.cygnus.com/cygwin/` so that you can run `/bin/sh` and `make`. If you are running Windows and do not want to install the Cygwin toolkit, then see Using Ptplot under Windows without Cygwin

Below are the installation instructions for Unix and Windows with Cygwin.

1. Be sure that JDK1.2 is installed, and that `javac` is in your path.
2. Untar or unzip the Ptplot distribution.
3. Set the `PTII` environment variable to point to the top of the Ptplot tree. Under C-shell, one would do:

   ```
   setenv PTII ~/src/ptplot3.1
   ```

4. cd into the `PTII` directory:

   ```
   cd $PTII
   ```

5. Run `configure`

   ```
   ./configure
   ```

6. Run make

   ```
   make
   ```

7. View the applets by going starting your browser and going to: ptolemy/plot/doc/index.htm

Installshield Installer

Ptplot is also available in several self-extracting formats that were created with the Installshield JShield tool.

Our use of this tool is still experimental. If you have problems, please try the tar or zip files.

Note that when you run the installer under Windows, Ptplot is added to your Start menu under Ptolemy -> Ptplot.

Installshield limitations

- When the Installer Applet is run under Windows, the Installer Window is not listed in the Windows Start Bar.
- The File browser does not work very well. Installshield says that this is because we are using JDK1.2. The standalone applications require JDK1.2 so that they can use Swing. Note that the applets can still be run with JDK1.1.x, so we don't require JDK1.2.
- Installing Ptplot in a directory whose pathname has a space in it seem to cause problems with the start up scripts.
- If you choose the Microsoft jvm, which sometimes found at `c:\winnt\jview`, then the format window will not come up. This is probably because the format window uses Swing facilities that are not in `jview`
- Ideally, Ptplot should be installed on machines with JDK1.2. It is best if you install Java before installing Ptplot. However The Installshield applet installer can download and install a JRE for you. Usually the JRE that can be downloaded via the Installshield applet installer is JDK1.1.7B. We were able to substitute in the JRE 1.2.2 .exe file so that under Windows, JRE1.2.2 is properly installed. However, we were not able to get this to work under Solaris, it appears that the JRE1.2.2 installation hangs part way through the installation. As a result, JRE1.1.7B is the JRE that is optionally downloaded for Sparcs.
- The InstallShield installer places the `ptplot`, `histogram` and `pxgraph` startup scripts in the toplevel `ptplot3.1` directory rather than placing them in `ptplot3.1/bin`

Notes about building and installing

How does configure work?

`configure` is a `/bin/sh` script that determines some information about your environment and then modifies a few files. When the `configure` script is run, it reads in `mk/ptII.mk.in` and `ptolemy/plot/ptplot.in` generates `mk/ptII.mk` and ptolemy/plot/ptplot

To get help with `configure`, try

```
./configure --help
```

`configure` reads in the `.in` files and substitutes strings inside the `@`

For example, `configure` reads in `ptolemy/plot/ptplot.in` and

```
PTII_DEFAULT=@PTII_DEFAULT@
PTJAVA_DIR=@PTJAVA_DIR@
```

becomes

```
PTII_DEFAULT=/users/cxh/ptII
PTJAVA_DIR=/usr/java1.1
```

Using Ptplot under Windows without Cygwin

Below are the instructions for using Ptplot if you are under Windows and you do not have the Cygwin and have downloaded the tar or zip files.

1. Set the `PTII` variable to the top-level directory of the Ptplot tree.

   ```
   C:\ptplot3.1> PTII=c:\ptplot3.1
   ```

   `PTII` is used by the standalone `ptplot.bat` script and other scripts
2. Be sure that `java` is in your path by running `java -version`
3. Added `%PTII\bin` to your PATH.
4. Run `ptplot.in`

Usually, it is not necessary to recompile the Ptplot classes, but below are the instructions for recompiling by hand under Windows without Cygwin.

1. To compile the Ptplot Java classes, they must be in the directory ptolemy\plot.

   ```
   C:\ptplot3.1> cd ptolemy\plot
   ```

2. set your CLASSPATH to ..\..

   ```
   C:\ptplot3.1\ptolemy\plot>set CLASSPATH=..\..
   C:\ptplot3.1\ptolemy\plot>echo %CLASSPATH%
   ..\..
   ```

3. Remove the old Java files and compile them

   ```
   C:\ptplot3.1\ptolemy\plot>del  *.java *\*.java
   C:\ptplot3.1\ptolemy\plot>javac *.java
   ```

4. Run `ptplot.bat`, which reads `%PTII%`

   ```
   C:\ptplot3.1\ptolemy\plot>ptplot.bat
   ```

Installing Ptplot as an applet

Ptplot includes several jar files, which are collections of Java `.class` files

`plotapplet.jar`
: .class files necessary for simple JDK1.1 browser applets - no pxgraph, no plotml.

`pxgraphapplet.jar`
: .class files necessary for simple JDK1.1 browser applets with pxgraph classes, no plotml. Applets that use the `pxgraphargs` applet parameter should use this jar file.

`plotmlapplet.jar`
: .class files necessary for PlotML applets - includes com/microstar/xml and the plotml code, but does not include pxgraph, or PlotFrame.

`plot.jar`
: The standard Ptolemy II jar file that includes the ptolemy/plot/*.class files and the class files in plotml and compat. Does not include com/microstar/xml or ptolemy/gui. `plot.jar` is used by the Ptolemy II build system.

`plotapplication.jar`

.class files necessary for standalone applications - includes com/microstar/xml, ptolemy/gui and the plotml code, but does not include pxgraph. If `plotapplication.jar` is present, then it is used by the standalone scripts (`ptplot` etc.)

The best way to use Ptplot as an applet is to copy the appropriate `*applet.jar` file into the same directory as your applet, and then to use the `archive` applet directive. For examples of applet html code, see the demonstrations.

Installing Ptplot as an application

Under Unix and Windows with the Cygwin toolkit, ptplot can be run as three standalone applications:

- `ptplot` - A modern plot application that with a menubar and such.
- `histogram` - A histogram plot application
- `pxgraph` - A plot application that is mostly backward compatible with the older X11 `pxgraph` program

When configure is run, it substitutes variables inside `ptplot.in` and produced `ptplot`. The `ptplot` script is shared between the standalone applications, `ptplot` determines what name it was called with, and selects the appropriate class accordingly.

If the `ptplot` script is run, then `ptolemy.plot.PlotApplication` is run

If the `ptplot` script is copied to `histogram`, then `ptolemy.plot.plotml.HistogramMLApplication` is run

If the `ptplot` script is copied to `pxgraph`, then `ptolemy.plot.compat.PxgraphApplication` is run

Under Unix, you can use a hard link to create the histogram and pxgraph script:

```
ln ptplot histogram
ln ptplot pxgraph
```

Under Windows, three .bat scripts are provided: `ptplot.bat`, `histogram.bat` and `pxgraph.bat`

Last Updated: $Date: 1999/08/20 15:58:52 $

Google

Web Images Groups News Froogle Local[New!] more »

JAR applet installer Search Advanced Search Preferences

Web

Results **1** - **10** of about **16,300** for **JAR applet installer**. (**0.18** seconds)

Sponsored Links

Java **Installer**
install4j creates installers and launchers for all major platforms.
www.install4j.com

Vitalsecurity.org - We're Calm like a Bomb: Firefox Spyware ...
... the Java Runtime Environment, the initial **installer** taking the ... used can recognise and install the **applet**, then it ... of you at the back, here's the .**Jar** file in ...
www.vitalsecurity.org/2005/ 03/firefox-spyware-infects-ie.html - 101k - Mar 19, 2005 - Cached - Similar pages

JNLP Tags - Quick Reference
... optional, Describes the documentbase for the **applet** as a ... class attribute if the first **jar** in the ... extension, Describes a component or **installer** extension that is ...
www.vamphq.com/jnlpquick.html - 29k - Cached - Similar pages

JNLP Tag Reference
... file as an application update - only updated **jar** files will. ... security?, resources*, (application-desc | **applet**-desc | component-desc | **installer**-desc) ...
www.vamphq.com/jnlpref.html - 83k - Cached - Similar pages
[More results from www.vamphq.com]

Web **Installer**
... If you wish to sign the **applet** with your ... Once signed, place the WebInst.**jar** and WebInst.cab ... InstallAnywhere/resource/Web folder, and rebuild your **installers**. ...
www.zerog.com/iamanual/usermanual_ia55/ WebHelp/distribution/web_**installer**.htm - 7k - Cached - Similar pages

Signing **Applets** for Java Plugin
... Security System and Using PVCj As Your **Installer** for the ... you have to somehow get the PVCj **jar** file onto ... modify your HTML page that starts the **applet** to point ...
www.suitable.com/docs/signingsignplug.html - 16k - Cached - Similar pages

JNI : Java Glossary
... The **installer Applet** will copy the Worker.**jar** to C:\program files\Netscape\Communicator\ program\java\classes or perhaps D:. Thereafter, you could run it ...
www.mindprod.com/jgloss/jni.html - 44k - Cached - Similar pages

Deploying Java Extensions in Java Plug-in
... The Java **installer** must be bundled as a **JAR** file, and the resulting **JAR** file must be specified as Implementation-URL in the **applet's** MANIFEST file. ...
java.sun.com/products/plugin/1.3/docs/extensions.html - 24k - Cached - Similar pages

JavaFoil - the **Applet**
... javafoil.**jar** and mhclasses.**jar**) into this directory. ... You can run the **applet** by loading this html page into ... you can also download a Windows **Installer** file which ...
www.mh-aerotools.de/airfoils/jf_**applet**.htm - 12k - Cached - Similar pages

Installing Ptplot
... The InstallShield **installer** places the ptplot , histogram and pxgraph startup scripts in ... Ptplot as an **applet** is to copy the appropriate ***applet.jar** file into ...
www.cs.bham.ac.uk/resources/ums/java/ packages/ptplot3.1/ptolemy/plot/doc/install.htm - 10k - Cached - Similar pages

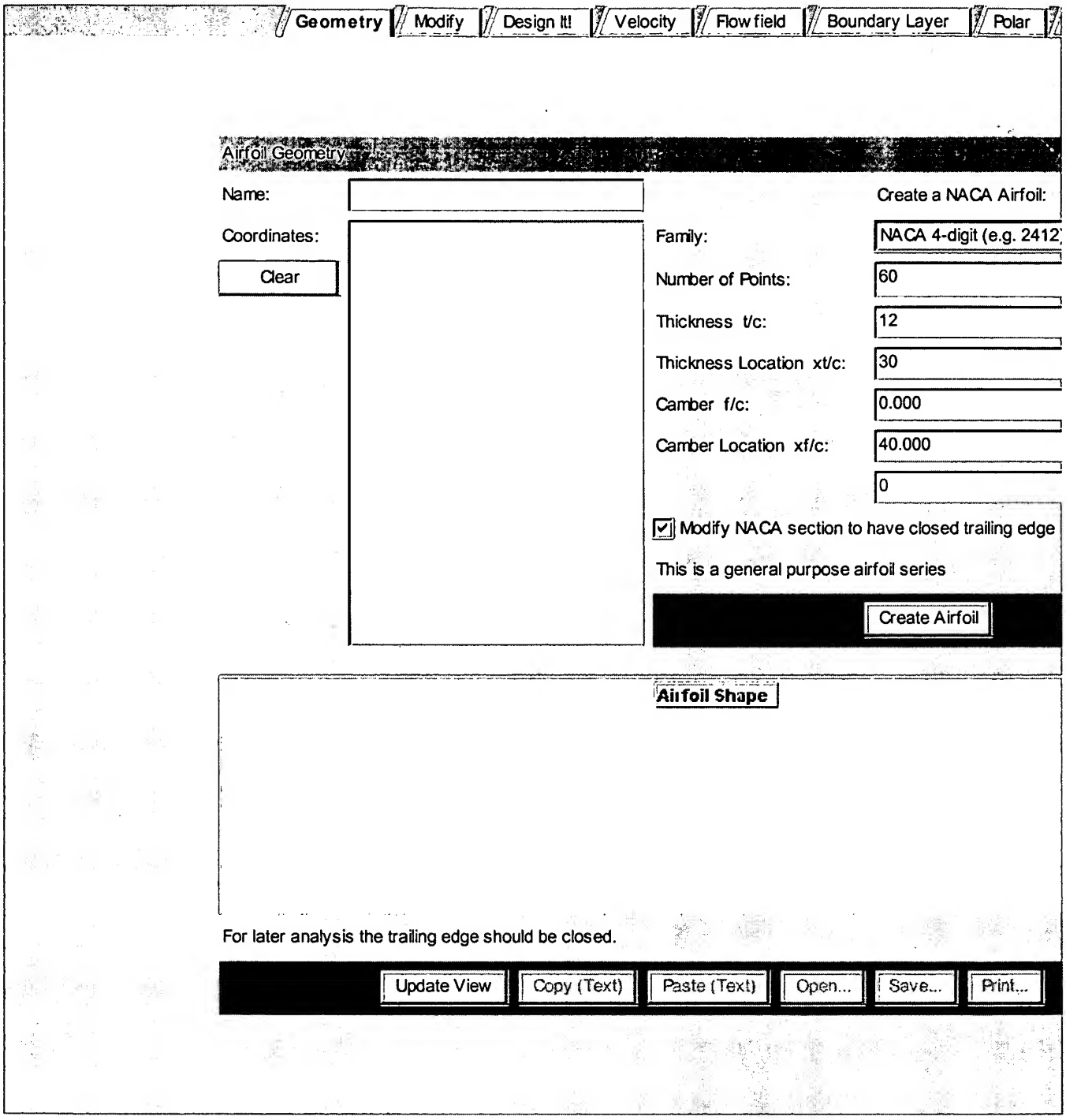
Install Autodesk MapGuide Viewer, Java Edition Release 6.5
... Choose "Uninstall" from the **applet installer**. OR; Open the folder: [Your User Home directory] > Library > Java > Extensions. Drag the mgjava.**jar** file to the ...
navigatela.lacity.org/download/Distribution/Install/ - 22k - Cached - Similar pages

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

JAR applet installer Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google



Step by Step Instructions to download the files required for a *local copy* of *JavaFoil*

- Check, whether you have already a Java virtual machine (VM) installed on your system.
  - Windows 98, 2000, XP: open a command window and execute the commands **jview** and **wjview**. Usually a Java VM can be installed during the installation or when upgrading the Microsoft Internet Explorer.
  - Unix: open a shell or xterm window and execute the commands **java** and **appletviewer**.
- If these commands are found, your system probably has a working Java installation. If not, you will have to get the appropriate Java Runtime Environment (JRE) for your system from Suns web site http://java.sun.com/.

- There seem to be problems with some older Java versions. You should have at least Java runtime version 1.1 installed.
- Download this html file from

  http://www.mh-aerotools.de/airfoils/jf_applet.htm

- Download the JAR archive with the class tree from the following web address:

  http://www.mh-aerotools.de/airfoils/java/javafoil.jar

- Download the JAR archive with the utility package class tree from the following web address:

  http://www.mh-aerotools.de/airfoils/java/mhclasses.jar

  This archive is also used by *JavaProp* - you have to download it only once.

- Create a subdirectory "java" below the directory, where you have copied this html file and move the files *javafoil.jar* and *mhclasses.jar* into this directory.

- Now you have both files to run *JavaFoil* either as an applet or a standalone application. You can run the applet by loading this html page into your browser, by using the Java AppletViewer or by running it from the command line.
  [More details can be found here].

- To make life for Microsoft Windows users even easier, you can also download a Windows Installer file which contains all *JavaFoil* files and installs them correctly.

  http://www.mh-aerotools.de/airfoils/java/javafoilinstaller.msi

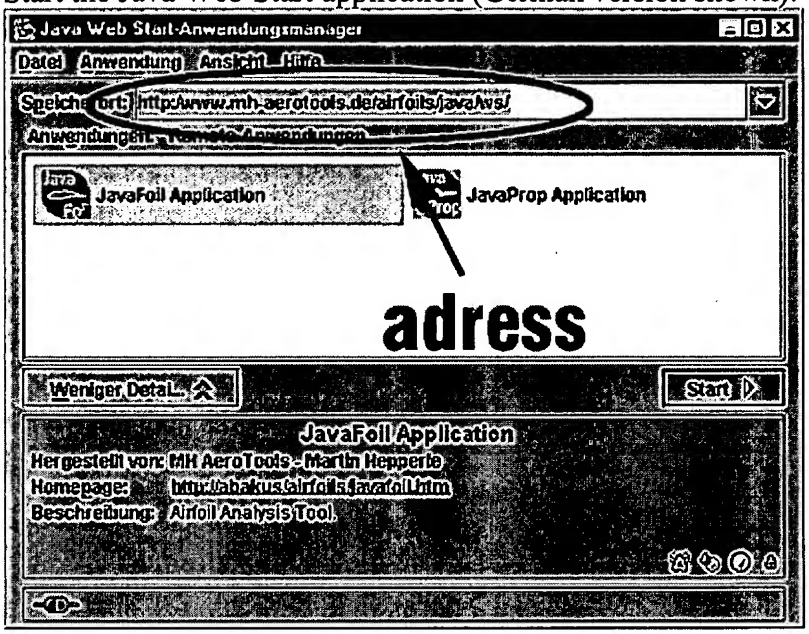
  As the direct download does not work as expected on some systems (you receive a "view" of the the file on screen), you should use the context menu (right mouse button) of your browser to download the .msi file (**M**icrosoft **I**nstaller).
  Then you can run the .msi file or use the context (right mouse button) menu to install.
  The Installer will install an html file to be used with your browser as well as a VBScript file (.vbs) which can be used to run *JavaFoil* without a browser by clicking onto the .vbs file. You can also de-install *JavaFoil* running the Installer again or by using the *Software* applet in the System Control program.

  The *Microsoft Installer* is not part of Windows 95 and 98 but can be downloaded from the Microsoft web site:

  - Redistributable v2.0 for Windows 95, Windows 98, and Windows ME:
    http://www.microsoft.com/downloads/release.asp?releaseid=32831&area=top&ordinal=19
  - Redistributable v2.0 for Windows NT 4.0 and Windows 2000:
    http://www.microsoft.com/downloads/release.asp?releaseid=32832&area=top&ordinal=8
  - Windows 2000 and Windows XP:
    the Installer is included with the basic system. If not, you can probably also use the NT 4.0 version above.

Instructions if you have *Java Web Start* installed (Java 1.4)

- If you want to install *Java Web Start* visit Suns web site at http://java.sun.com/products/javawebstart/ for availability of Java Web Start downloads.
- Start the Java Web Start application (German version shown):



- Enter the address
  http://www.mh-aerotools.de/airfoils/java/ws/remoteapps.htm
  into the address field and press the RETURN or the TAB key to see all available applications.
- The Web Start application will scan the address and show you all available applications.
- When you start an application for the first time, you are asked whether you trust it and allow access to files, printer etc. If you grant access, you can run it like a normal application.
- You can later work offline, because Java Web Start will cache the application files.
- For Java 1.5 see **http://www.mh-aerotools.de/airfoils/java/ws/remoteapps.htm**

Last modification of this page: 15.03.05

[Back to Home Page] **Suggestions? Corrections? Remarks? e-mail: Martin Hepperle.**

Due to the increasing amount of SPAM mail, I have to change this e-Mail address regularly. You will always find the latest version in the footer of all my pages.

It might take some time until you receive an answer and in some cases you may even receive no answer at all. I apologize for this, but my spare time is limited. If you have not lost patience, you might want to send me a copy of your e-mail after a month or so.

This is a privately owned, non-profit page of purely educational purpose. Any statements may be incorrect and unsuitable for practical usage. I cannot take any responsibility for actions you perform based on data, assumptions, calculations etc. taken from this web page.

Component Java

The Applet is dead; long live the Applet!

Michael-Franz Mannion

Michael-Franz Mannion is CEO of the Swiss-based IT consulting firm, Cutting Edge GmbH. He can be contacted at michael-franz. mannion@ubs.com.

This article provides an overview and introduction to Java Web Start, an exciting new Java application distribution technology and reference implementation of the Java Network Launching Protocol (JNLP).

I've heard it said that Sun never really intended applets to be positioned as a solution to automated application distribution, but more as a means to market the Java platform. To be fair, applets and Java technology did (even from the outset) address major portability and security issues by providing platform-independent byte-codes, byte-code verification, and sandboxing; but several critical issues remained unaddressed.

Enter JNLP, aka JSR-56, a technology that aims to provide a complete solution to the problem of Java app distribution.

Sun provides a free reference implementation of JNLP, called Web Start, which enables desktop PCs and workstations to download Java applications with little or no involvement from the end user. Of course, Web Start is just one possible implementation of JNLP. For example, a JNLP client written for a Sony Playstation is unlikely to implement JNLP in the same way as Web Start.

Web Start Features

Here's a summary of Web Start's features:

- Fully automated, Web-centric distribution and installation of Java 2 applications, applets, and extensions based on the JNLP;
- Resource caching—Application components are cached automatically on the client's machine;
- Browser independence—Applications are executed outside of the browser process and can also be launched directly from the desktop;
- JVM independence—A pre-requisite virtual machine implementation and version can be specified and, if not already present on the client machine, downloaded and installed automatically;
- Transparent updating—Versions of cached application resources are checked against those hosted on the Web server. Newer versions are downloaded and installed automatically;
- Incremental updates—Only new or modified classes and resources need be uploaded to the client's machine; and
- Incremental downloads—If required, archives can be downloaded only when first required as opposed to being downloaded immediately.

Let's make it clear that Web Start is *not* applets 2.0. It has nothing directly to do with the `java.awt.applet` API, although, as we shall see, the critical sandboxing features of applet technology have been retained and enhanced.

Application Life Cycle Under Web Start

Here's how an application makes it to a client's machine under Web Start:

1. Using a browser, the end user clicks on a JNLP file—an XML document describing the application—which causes the file to be downloaded to the client.
2. Web Start is started (by the browser) and presented with the downloaded file.
3. If the JNLP file describes an application's resources that are not already present on the client's machine, these resources will be downloaded using the specified URL information in the file.
4. Once downloaded, the application is started (in its own process) using program parameters obtained from the JNLP file.

The second time a JNLP file is referred to, the situation changes somewhat in Step 3: The Application Manager recognizes that the application is already present and compares the newly downloaded JNLP file with an old file. If it is established that a newer version of the application exists, the updated resource(s) will be downloaded accordingly. If not, no download takes place.

The sequence in Figure 1 shows the logical interactions involved for a newly requested application. This application comprises a single resource called `wsexample.jar`. Fortunately for the end user, the only thing to be done is to click once on the hyperlink that points to the JNLP file.

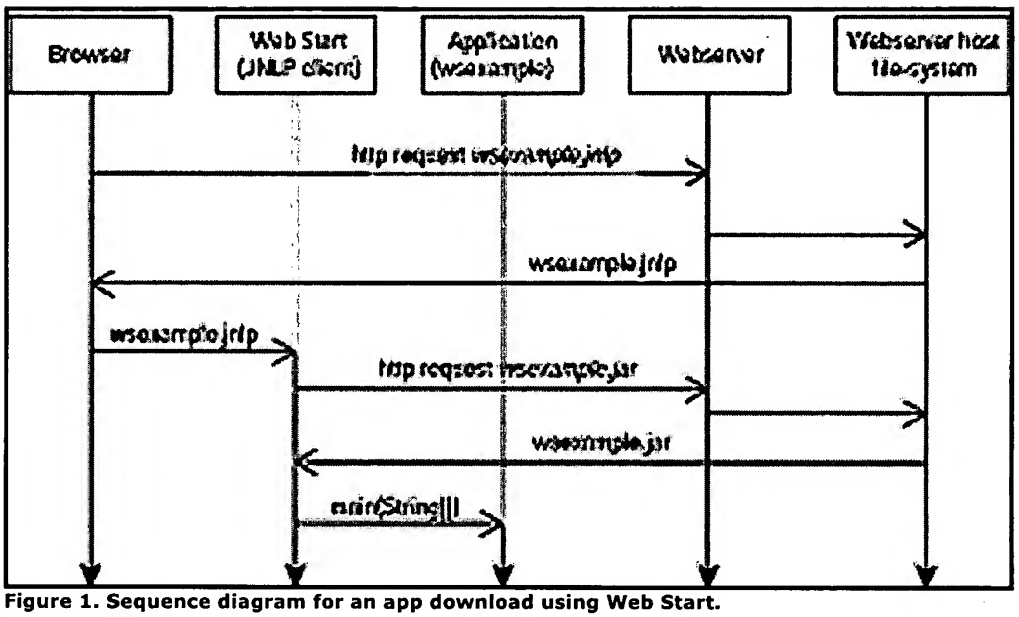


Figure 1. Sequence diagram for an app download using Web Start.

Installing Web Start

There is a bootstrapping procedure associated with automated application distribution, and it begins with the installation of Web Start on the client machine. Installation requires the following steps:

1. Download Web Start for your chosen operating system from the JavaSoft Web site. (Version 1.0.1 includes JRE 1.3.)
2. Run the installation program. The installation program adds the MIME type `application/x-java-jnlp-file` to the browser's MIME type database in association with the file extension `jnlp`. Exit the browser if prompted to do so.

To start the Application Manager under Windows, either double click on the newly installed icon on the desktop or double click on `\javaws.exe`. Under Unix/Linux, go to the install directory and type `./javaws`. The Application Manager window should now appear (see Figure 2).

Figure 2. Initial start-up of Web Start's Application Manager.

Configuring and Testing Web Start

That's pretty well it for the client installation, except that for those wishing to access applications hosted beyond a firewall, Web Start's proxy server settings will need to be specified. To do this:

1. Start the Application Manager.
2. Go to File -> Preferences.
3. Depending on how you usually specify proxy settings, either click on "use browser," or click on "manual" and enter the URL (omit the prefix "`http://`" as it will not work) and port of the proxy server, which you can obtain from your network admin (see Figure 3).

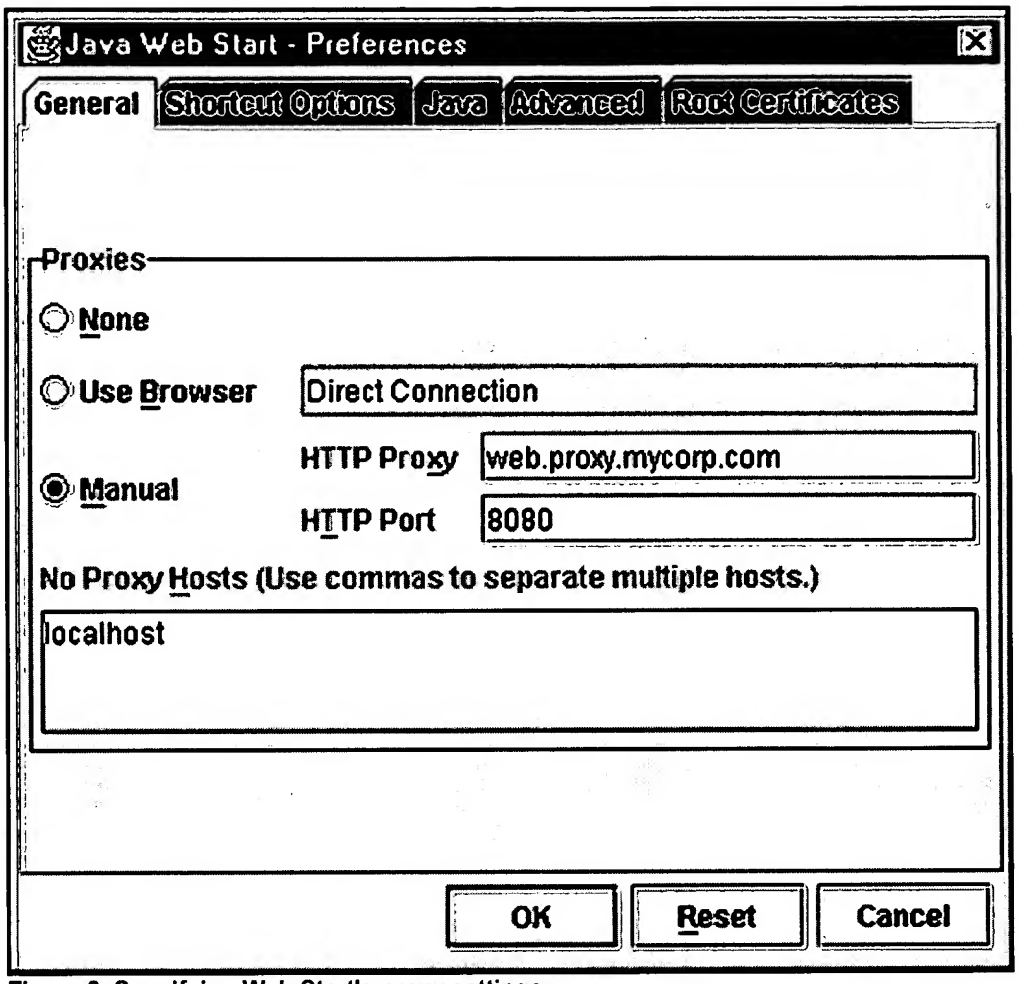


Figure 3. Specifying Web Start's proxy settings.
When using Web Start for the first time, as well as for debugging, it is worth selecting the console and logging options to "on," together with a filename for the log file. These options can be found under File -> Preferences -> Advanced (see Figure 4).

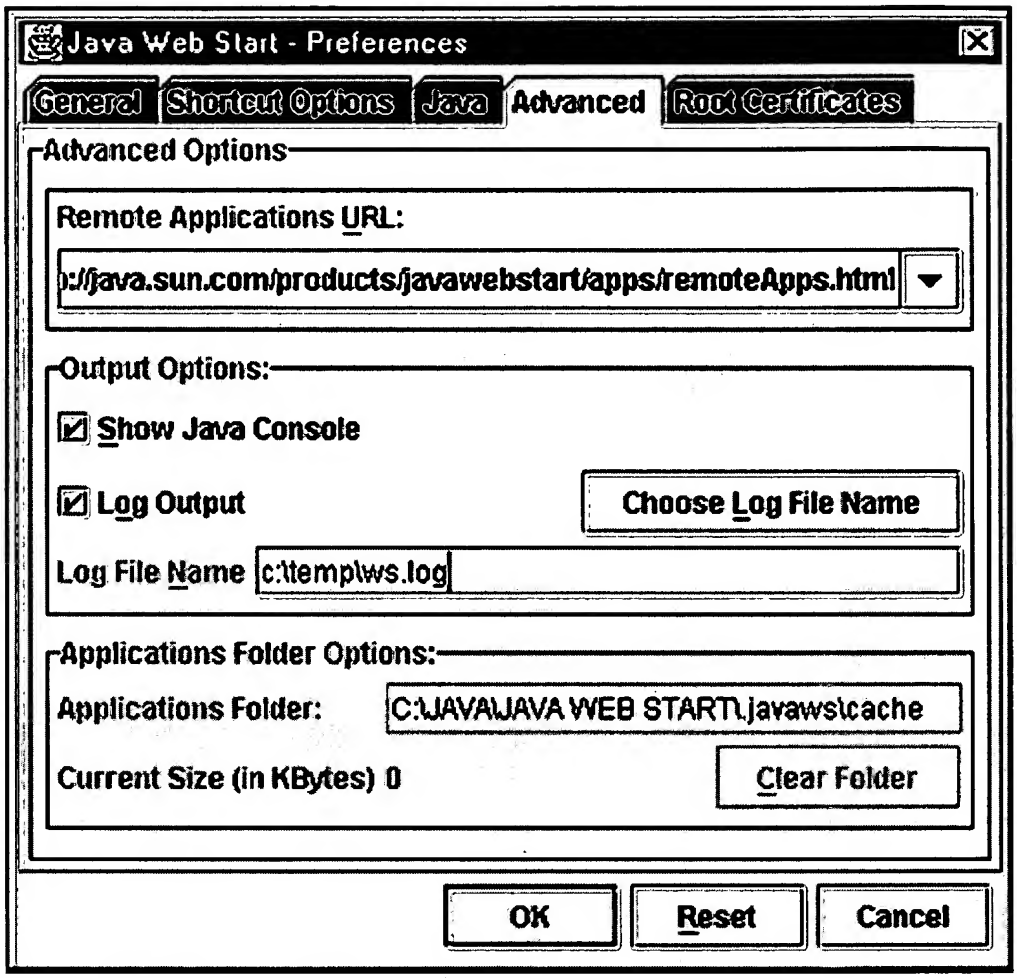


Figure 4. Specifying console/logging options.

On the advanced tab, we can find another important feature: Clear Folder. This function is used periodically to clear cached applications from the client's system. A consequence of its usage is that apps need to be downloaded once again.

You can now verify if everything worked by trying out the demonstrations at the Web Start demo site. To run the demonstration, simply click on an app icon that downloads (via HTTP) the app's JNLP file. Web Start's Readme file provides some troubleshooting info should things not work as expected.

Once downloaded, the now cached application can be re-launched from the Application Manager (see Figure 5).

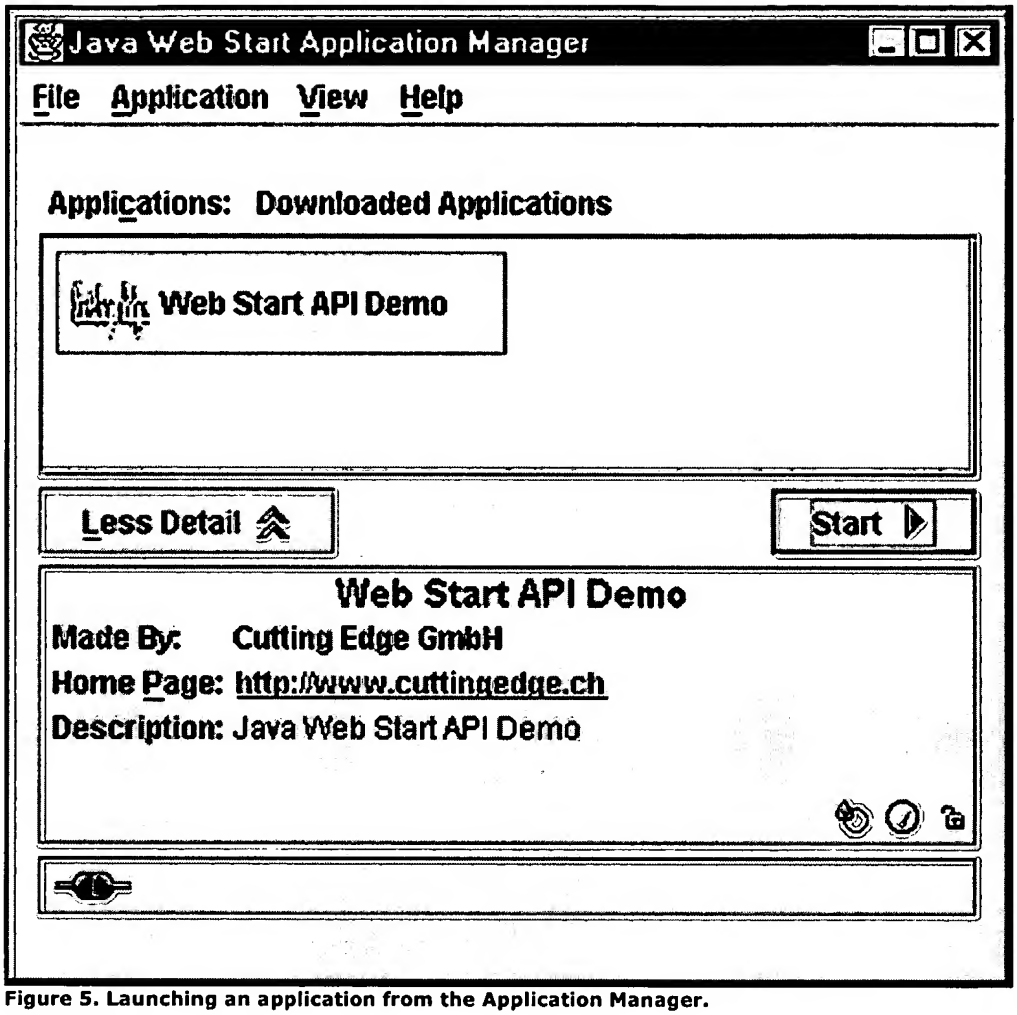


Figure 5. Launching an application from the Application Manager.

The JNLP File

The JNLP file is an XML descriptor for the application, its resources, and its security requirements. Every JNLP application needs a JNLP file (see Listing 1).

The top-level element of the JNLP file is called . The "codebase" attribute defines the base address of application resources, whereas the "href" attribute gives the location of the JNLP file itself. In general, for attributes that have URLs as values, it is possible to specify either a relative URL value (relative to "codebase," that is) or an absolute value. An absolute value will include the protocol prefix `http://`.

Application Descriptor Elements

The element of Listing 1 indicates that the JNLP file represents an app. The attribute "main-class" informs Web Start where the program's `public static void main()` method can be found, for example:

```
:

    starlet
    300

:
```

Program arguments can also be specified by adding the sub-elements. These end up in the string array argument of the `main()` method.

Applets Descriptor Elements

Applets can also be deployed with JNLP by providing an as opposed to an . The attributes of this element correspond to those we are used to providing in HTML using the applet tag, i.e., applet class name, document base, width, height, and parameters.

```
    :
```

:

:

Extension Descriptor Elements
Then there are the so-called "extension elements" and . A element describes a component extension that typically groups resources common to several apps; for example, frequently used GUI or back-end connectivity components. The primary benefit of this is that you can version and then deploy these common components independently of the applications. A component extension has no attributes or sub-elements and so can be written thus:

:

:

Finally, an element describes a so-called installer extension, which is primarily used to install software elements that require a platform-dependent setup, such as a JRE or device driver. Like the application descriptor, the installer descriptor must also name the software's entry point by specifying a class with a `main()` in it:

:

:

Installers are run only once for any given client. Extension descriptors serve to modularize the various resources of a rollout. The complete description of two non-trivial applications could therefore be structured as in Figure 6.

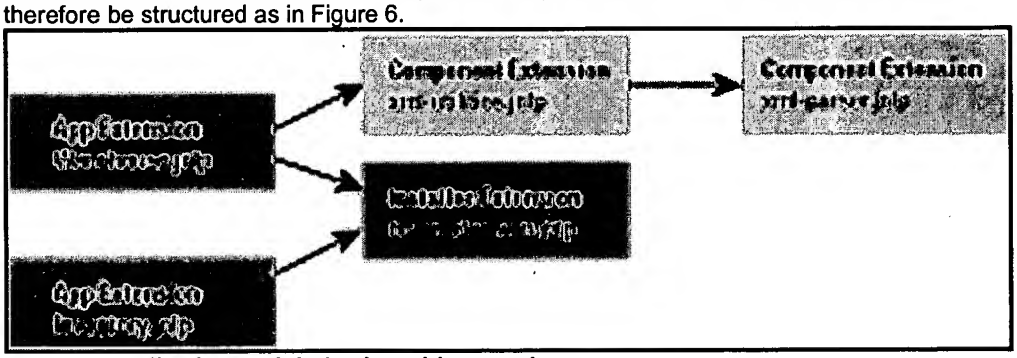


Figure 6. Application modularization with extensions.

Resource Elements
Using elements, we specify the jars, native libraries, pre-requisite JRE, properties packages, and extensions (component or installer) that make up the application.

The resources element has some optional attributes, two of which are `os` and `arch`. By specifying this, we can state that a resource is designed for a particular operating system and architecture only, for example:

This is obviously of particular significance to native libraries. The values presented correspond to the system properties `os.name` and `os.arch`, respectively.

Resources can be downloaded either immediately or on demand using the download attribute, with values "eager" or "lazy," respectively. The default here is "eager." This feature can be used to reduce start-up latency of an app.

:

:

:

:

Finally, a resources element may refer to component or installer extensions.

:

href="libs/bc-reader-win.jnlp">

:

:

The JNLP files referred to here are downloaded and processed in much the same way as the parent JNLP file. If the addressed extension is a component extension, then the extension's resources will be downloaded and will ultimately form part of the app's CLASSPATH. If it points to an installer extension, then the installer will be downloaded and executed.

The Security Element

The element of the JNLP file describes the permissions environment in which the application needs to operate. Omitting the security element altogether gives the application similar rights an unsigned applet has when downloaded by a browser:

- All jar file resources named in the element of the JNLP file must originate from the same host; and
- No native libraries can be used, i.e., no nativelib elements can be specified in the JNLP.

In contrast to an applet, an unsigned JNLP application is permitted to exit the JVM (with System.exit()).

For trusted applications, the current JNLP spec currently names two possible, mutually exclusive sub-elements of the security element, which look like this in the JNLP file:

:

:

or

:

:

The element refers to the permissions environment java.security.AllPermissions, which grants the app the same level of control over its execution environment as a regular

standalone application. The refers to a limited permissions environment that, among other things, allows socket connections to be opened to an arbitrary host and to load native libraries. For the full list of permissions for these variants, refer to the JNLP spec at JavaSoft's Web Start site.

The following conditions must, however, be satisfied before a launched Web Start application is granted these permissions:

1. The application is signed; and
2. The end user trusts the certificate used to sign the application. To this end, Web Start prompts the end user to accept or reject the certificate.

The Information Element

The JNLP file's element provides meta-information about the application. This information is typically shown to an end user during downloading or following downloading in the window of the application manager (see Figure 5).

```
  Cutting Edge GmbH

   Corporate Inventory


   No!->
```

One can also specify for the program to run offline by using the sub-element. In this example, however, this is not the case.

Making Your Own JNLP-Enabled Applications

Any Java program, including those that depend on native code, can potentially be deployed via JNLP. Here's what to do:

1. Ensure your application can access resources (icons, properties files, etc.) from a jar file (see "Accessing Program Resources" below).
2. Package your application and other resources in one or more Java aARchive (jar) files. If the application requires privileged access to client resources, it will have to be signed.
3. Write a JNLP file for the application by copying and then modifying an existing JNLP file, such as the one in Listing 1.

Accessing Program Resources

One of the rules of packaging JNLP applications is that everything must be packaged in a jar file. This rule also applies to program elements such as icons and properties files, etc., which—before Web Start—one might have been tempted to obtain from the file system using a `FileInputStream`.

To access these items from a jar, we must go via a class loader, as the following code excerpt demonstrates:

```
:
ClassLoader cl =
       getClass().getClassLoader();
Icon corporateLogo = new ImageIcon(
       cl.getResource("images/logo.gif") );
:
```

This code assumes a jar structure as follows:

```
:
com\mycorp\code\MainProg.class
:
images\logo.gif
:
```

This is a common but, thankfully, simple software modification that must be executed when making existing applications JNLP-enabled.

Setting Up a JNLP Application Host

JNLP applications need to be hosted on an HTTP server. Fortunately, this is usually quite easy:

1. Set up the Web server so it associates JNLP files with the MIME type `application/x-java-jnlp-file`. With Tomcat, for example, make a MIME mapping entry in the file `Tomcat/conf/web.xml`.
2. Make the application's resources (jars, etc.) and JNLP file accessible via HTTP by placing them in the Web server's document path. The URL of the application specified in the JNLP file must obviously correspond to the jar's location.
3. Provide a link from some Web page to the JNLP file.

For the most basic behavior, these few steps are all that is required. Of course, with this basic setup, providing an application update implies producing a new jar and the client uploading—even if 90% of the application is unchanged. Fortunately, Web Start allows us to be much more economical with system resources than this.

Supporting Incremental Updates

Incremental updates permit a program's updated parts (its "deltas") to be shipped and installed, rather than the entire application. This can significantly increase updating speed, as well as reduce network traffic.

Incremental updates come into play when an exact or, alternatively, a minimum version number is specified for a particular resource in the JNLP file, for example:

When a JNLP client, such as Web Start, comes to request this resource from the HTTP server, it must encode the HTTP GET request according to the JNLP specification, which incorporates the version id of the requested resource.

```
http://www.cuttingedge.ch/wsapps/wsexample.jar?
version-id=1.1%2B¤t-version-id=1.0
```

In this example, we see the JNLP client has also stated the *current* version id. This extra information can be used by the application host to ship a software delta. But to support this feature, the Web server needs to be equipped with a little intelligence and this is typically provided by a servlet.

Version 1.0.1 provides the `JnlpDownloadServlet`, which serves exactly this purpose. In fact, the `JnlpDownloadServlet` does a lot more than deliver software updates. First, the servlet provides a convenient mechanism for deploying application resources from a Servlets 2.2 capable servlet container (e.g., Apache's Tomcat) by allowing the software to be packaged as a Web application archive or war file. Here's an example war file structure that corresponds to the JNLP in Listing 1:

```
/index.html
/app/warehouse.jnlp
/app/warehouse_V1.1.jar
/app/images/logo.gif
/WEB-INF/web.xml
/WEB-INF/lib/jnlp-servlet.jar
/WEB-INF/lib/jaxp.jar
/WEB-INF/lib/parser.jar
```

We see that the `JnlpDownloadServlet`, as well as a JAXP-capable XML parser, are included in the war file. This and other details of setting up JNLP archives are explained in the guide that accompanies the developer's kit.

The `JnlpDownloadServlet` also automatically generates jar files that contain the software deltas on the fly by using the jardiff utility. This tool compares the previous version's jar with the newest version and creates a file representing the difference between them. The result, called a jar-diff file, is delivered to the client instead of an entire archive, whereupon the original jar is updated.

Using the `JnlpDownloadServlet`, there are two ways to specify the version numbers of resource archives. The simplest variation uses a special naming convention for the name of the archive. The war file structure above simply specified a version of V1.1. However, the name can also include operating system, architecture, and locale. For example:

```
:
/app/wsexample_V1.1_OWindows_NT_Ax86_Len.jar
:
```

```
where
V=Version
O=Operating System
A=Architecture
L=Locale
```

The operating system and architecture values here correspond to the attributes `os` and `arch` of the JNLP file's resource element.

The second way to specify resource versions is to provide an XML document, `version.xml`, that contains the required version information. Details of this are provided with the developer download. Whichever way you prefer will depend on your build procedures.

Finally, you need to make sure your servlet container is correctly configured so JNLP client's URL requests are sent to the `JnlpDownloadServlet`. Your servlet container's documentation tells you how to do this. But be aware that not all of today's servlet containers fully support JNLP, so consult your vendor before investing too much time.

Signing Applications

Since Java 2, the standard JDK comes with two command line tools that are used to sign each resource that makes up the application: `keytool` and `jarsigner`. The `keytool` utility is designed to manage a database of key and certificate information called a `keystore`. The `jarsigner` tool is used to electronically sign to a jar. Both of these tools come with JDK 1.2 and higher. The JDK1.3 documentation set comes with an improved user guide.

For test purposes, a test `keystore` can be generated using the following `keytool` command: `$ keytool -genkey -keystore testKeystore -alias mycerts`.

The command creates the `keystore` after you have entered some organizational information. To create a self-signed test certificate, use a command such as: `$ keytool -selfcert -alias mycerts -keystore testKeystore`.

Once the test certificate has been generated, an application resource `jar` is signed using the command: `$ jarsigner -keystore testKeyStore myComponent.jar mycerts`.

Of course, an application signed with a test certificate can hardly be called authenticated. "Credible" certificates are obtained from a reputable certification authority, either internally in some organizations, otherwise externally from a company such as VeriSign.

Once a signed application is downloaded to the client, the end user is prompted to either accept or reject the certificate (see Figure 7).

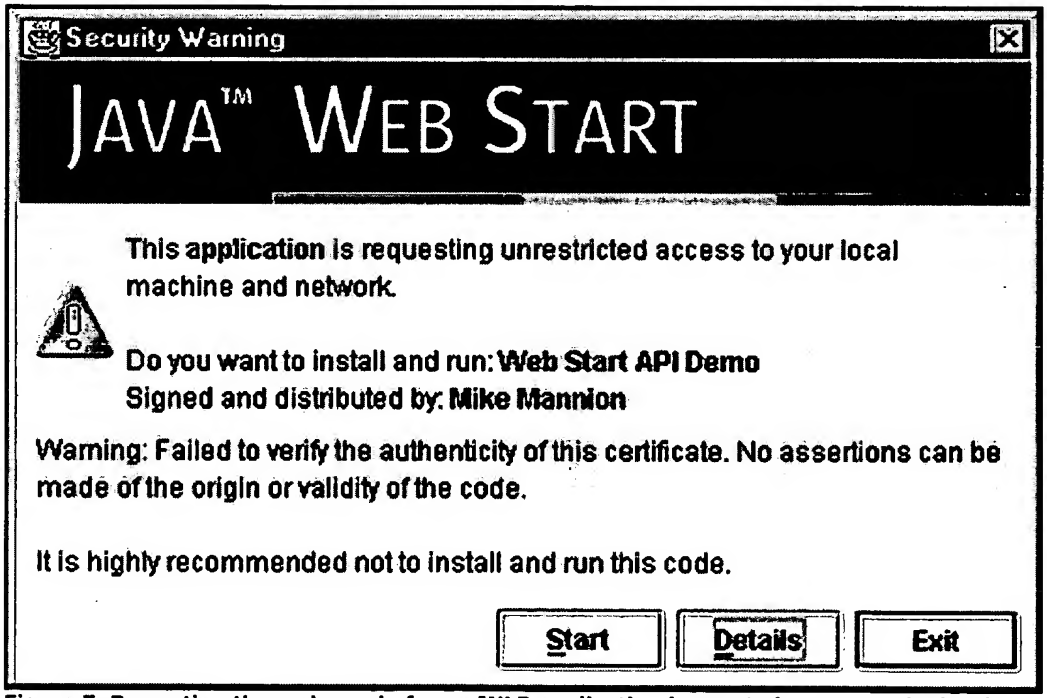


Figure 7. Prompting the end user before a JNLP application is granted access to the local machine.

The JNLP API

The JNLP API provides a number of services that give controlled access to resources not normally available to unsigned J2SE applications. Here's a summary of the JNLP services provided with Web Start 1.0.1:

- `BasicService`—Is similar to the `AppletContext` provided with the applets API. For example, the `showDocument()` method allows you to display the page for a given URL in the browser.

- `ClipboardService`—Gives the application access to the system clipboard. It also makes it possible to perform cut-and-paste operations involving other (non-Java) applications on the desktop.
- `DownloadService`—Gives the application access to some of the functionality any JNLP client itself requires. With it, you can invoke resource downloads and check to see if resources are already cached.
- `FileOpenService`—Permits untrusted applications to open streams to files on the local disk.
- `FileSaveService`—Allows untrusted applications to save files to the local disk.
- `PrintService`—Allows a local printer peripheral to be exploited. The functionality from the `java.awt.print` package is used to create the objects, which represent the printed medium.
- `PersistenceService`—Provides cookie-like functionality for the application. Note that it is not a general persistence mechanism and the amount of name-value pair data expected to be stored on the client machine is small. Amusingly, the designers have chosen to dub the underlying mechanism "`muffins`."

From a security perspective, several of these services could potentially give an application opportunity to reek havoc on a client machine. Fortunately, Web Start's implementation of these services includes checks with the Security Manager. So in cases where an application's requested permissions environment does not grant immediate access, the end user will be prompted to accept or reject the proposed operation (see Figure 8).

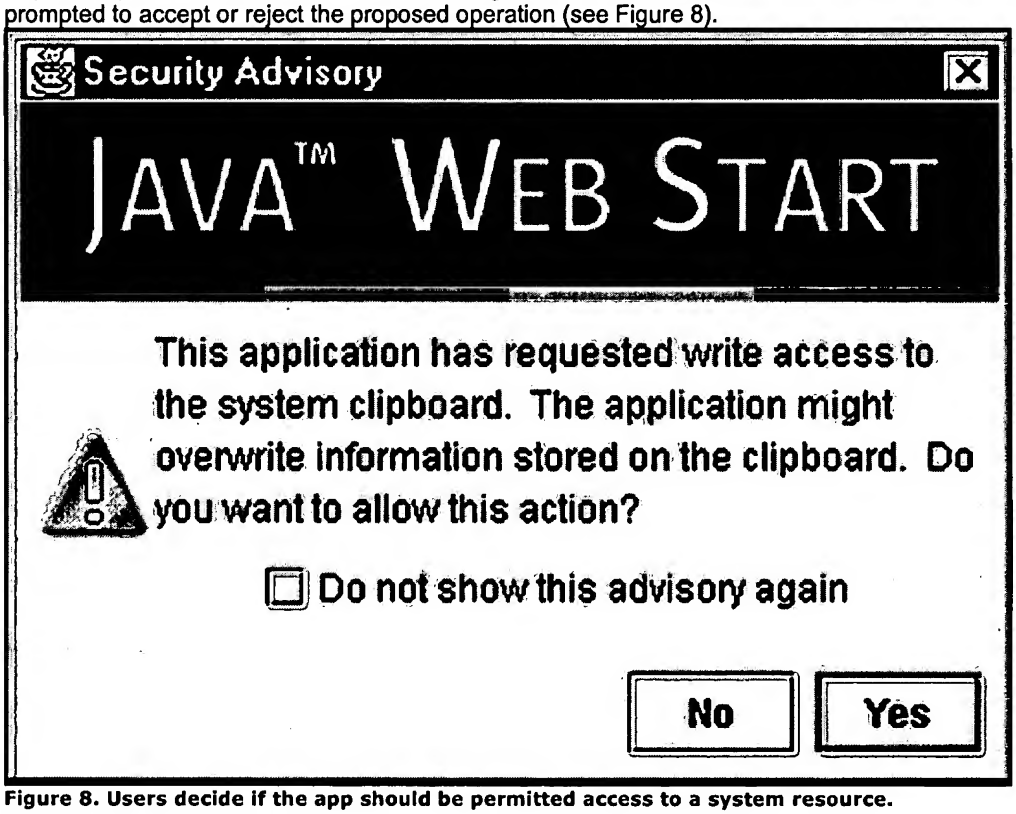


Figure 8. Users decide if the app should be permitted access to a system resource.

Conclusion

Perhaps partly because applets failed to fully address many issues of application deployment, there has been a trend over the last couple of years to develop every conceivable kind of Web application using Java servlets and HTML-related technologies. (In fact, some IT managers are still suffering the effects of applet shellshock, and will, therefore, continue to push HTML as the only viable solution.)

While this may still be justified for Internet applications, where the target audience is often unknown or application usage is infrequent, there now appears to be fewer reasons to continue with this style of application in intranet environments. This is especially the case when every end user has access to a standardized corporate desktop PC, which can include Web Start as part of the standard rollout. The CPU and memory resources of these boxes are simply *begging* to be used.

With Web Start's imminent integration into JDK1.4, along with 1.4's other promised major improvements in the client programming area, we shouldn't be surprised to see a big revival in Swing-based client development.

URLs

Web Start Download
http://www.javasoft.com/products/javawebstart/ index.html
Web Start Demo
http://www.javasoft.com/products/javawebstart/ demos.html
Developer's Kit

http://www.javasoft.com/products/javawebstart/_developers.html

VeriSign
http://www.verisign.com

JNLP Services Demo
http://www.cuttingedge.ch

Apache Software Foundation
http://www.apache.org

Installation - Download - Bugs & Features - Release Notes

Latest News

- Jan 13, 1999 JZipper support forum opened
- Jan 1, 1999 JZipper version 1.08 released, fixing issues with JDK 1.2
- Jan 1, 1999 JZipper Prerelase Jan 99 released
    - Fixed JDK 1.2 issues
    - Fixed Adjust signatures feature
    - Fixed Beans identification problem
    - .. some other bugs ..
- JZipper for JDK 1.02 is discontinued
- Nov 1, 1998 JZipper gets new owner, Vega Technologies Inc

General description

---**JZipper** is a certified 100% Pure Java application that eases the tasks associated with the publishing of Java Applets and Applications. JZipper allows the developer to specify the Java(TM) class files that are to be published and then it does the grunt work for you: zipping them together, optionally removing debug info and/or obfuscating symbolic names as it goes. The obfuscation feature makes it harder to de-compile your class files, protecting your work from the efforts of hackers trying to reverse engineer your code. Users can specify a main class and let JZipper include all dependant classes in the target ZIP or JAR file - dependant classes can be pulled right from other ZIP or JAR files! JZipper also converts ZIP files to JAR files effortlessly with the click of a mouse. Automatic removal of debug info eliminates the need for an extra release build and makes class files smaller reducing load time. JZipper provides both a convenient graphical interface and a handy command line feature enabling JZipper to be easily integrated with custom build procedures!

Exciting Features

- Specify a main class and let JZipper include all dependant classes in the target zip or jar file
- Dependant classes can be pulled right from other ZIP or JAR files
- Removes debug info from class files, eliminating the need for an extra release build
- Can obfuscate class methods and class names on the fly
- One can create config files which let JZipper to run on command line. This allows to integrate zip or jar file creation with build procedure

Technical information

---**JZipper** is written entirely in Java(TM) and the version for JDK 1.1 is certified 100% pure Java(TM). Consequently JZipper works on any platform where Java runs. It requires Java(TM) Development Kit (JDK) or Java(TM) Runtime Environment (JRE) installed. Both can be obtained from Sun Microsystems JavaSoft site. Depending on the version of JDK you have installed (1.0.2 or 1.1) there is a different version of JZipper for either JDK version. The 1.1 version also supports JAR files, one can actually convert zip files to JAR files with the ease of the mouse click. JZipper provides convenient user interface to select all the desired options and it provides command line processing feature allowing easy integration with build procedures. JZipper screenshot can be viewed here.

JZipper prerelease

---We are adding more exciting features to JZipper, some of them are already implemented, here is a list of newly added features:

- Automatic signing of the JAR file. This supports currently only JDK1.1, JDK1.2 implementation is forthcoming
- Cancel feature. Processing of all the classes can sometimes take a while, previously this couldn't be cancelled
- Memory consumption indicators
- Capability to specify multiple main classes
- Capability to browse classes in the CLASSPATH
- Automatic identification of Beans and inclusion of corresponding information into manifest file
- Mocha Death feature added to the obfuscation implementation
- Obfuscation enhancement: select between legal names and illegal names

- Obfuscation enhancement: define the prefix for the legal names
- Obfuscation enhancement: native methods are automatically recognized and not obfuscated
- Obfuscation enhancement: defined method names can be excluded from obfuscation. This feature allows wildcards, for example: one can define all the methods starting with get to be excluded from obfuscation
- Obfuscation enhancement: method signatures can be adjusted to the obfuscated names. This makes obfuscated classes compliant with JDK 1.2

---Would you like to see how the prerelease looks like? Click here for screenshot.

---Are you interested in trying the prerelease?

---Yes, I want to try out new prerelease-

Usage Examples

- You have just finished the development of an applet and intend to publish it. Since the applet involves multiple class files and also some classes from third party libraries you intend to publish it as a zip file. You fire up JZipper, browse to find the main class of your applet, usually the class which extends the java.applet.Applet, specify a target zip file name and press the Generate button. JZipper scans for all the dependencies of the specified main class and includes all the dependent classes in the target zip file. The dependent classes can be either individual class files or class files embedded in the zip file. Yes, indeed, JZipper can include classes right from other zip files. As an added bonus JZipper can obfuscate the classes, which makes decompilation useless. This way nobody can reuse your code without license
- You have just finished a development of a Java class library. Now it is time to publish it and the preferred way is to have it published as a zip file. Again you fire up JZipper, select a directory (package) and/or class files, specify a target zip file and press Generate button. JZipper includes all the selected class files and scans the selected directories for class files

---JZipper retails for $35 and can be purchased by navigating to one of the sites below:

- Netsales

---A fully functional free demo version can be downloaded here. Demo version will expire by the 15th of next month, by which time we hope you are ready to buy JZipper, however, you may always return to this site and download another demo version.

---Yes, I would like to try the JZipper demo-